Journal of
Software Engineering

ISSN 1819-4311

Academic
Journals Inc.

www.academicjournals.com

OPEN ACCESS Journal of Software Engineering

ISSN 1819-4311
DOI: 10.3923/jse.2017.40.46

Research Article
Optimization Model Based Web Application Distribution in Mixed Internet of Things

[1]Lei Wei-Jun, [2]Hu Ruo and [3]Gu Xiao Jun

[1]School of Mathematics and Computer Engineering, Xi-an University, 710065 Xi-an City, China
[2]Guangdong Polytechnic Normal University, 510665 Guangzhou, China
[3]School of Mathematics and Computer Science, Ningxia University, 750021 Yin-chuan City, China

Abstract
Background and Objective: This study was supported by the Xi-an Science and Technology Plan Project and Natural Science Fund Project in Guang Dong province. The purpose of the study is to present an optimization method that searches for an optimal solution to the exploiting weakness problem. **Materials and Methods:** The proposed optimization method was performed to determine the minimum number and the best positions of the moving web applications. Hence, the main finding is that moving web applications are added after the original distribution to overcome the exploiting weakness problem. To achieve optimal exploiting, a main method was employed to find the best positions of the extra moving web applications. **Results:** The results shows the exploiting ratio increases as the number of deployed non-moving web applications increases. **Conclusion:** The non-moving web applications are freely deployed after adding the moving web applications to the network. The performance of the optimization model was estimated using several methods and the simulation results showed that the proposed models can optimize the network exploiting in terms of the overall exploiting ratio and the numbers of extra moving web applications.

Key words: Object exploiting, web application, optimization model, internet of things

Received: May 13, 2016 Accepted: August 16, 2016 Published: December 15, 2016

Citation: Lei Wei-Jun, Hu Ruo and Gu Xiao Jun, 2017. Optimization model based web application distribution in mixed internet of things. J. Software Eng., 11: 40-46.

Corresponding Author: Lei Wei-Jun, School of Mathematics and Computer Engineering, Xi-an University, 710065 Xi-an City, China

Competing Interest: The authors have declared that no competing interest exists.

Data Availability: All relevant data are within the paper and its supporting information files.

INTRODUCTION

The exploiting issue in IOT depends on many factors, such as the network topology, node sensing model and the most important one is the distribution strategy that is used to distribute or throw the web applications in Agile software field[1]. The web applications can be deployed either manually based on a pre-defined design of the node positions or freely by dropping them from an aircraft. Free distribution is usually preferred in large scale IOT not only because it is easy and less expensive but also because it might be the only choice in remote and hostile environments. However, free distribution of the web applications can cause weakness formulation, therefore, in most cases, free distribution is not guaranteed to be efficient for achieving the required objective in terms of the exploiting web projects[2].

Depending on the required application, the web applications are responsible for sensing, computation and communication tasks. The sensing task is usually configured in each web application, therefore, the sensing attribute is considered a key factor in designing IOT. An Internet of Thing (IOT) is a distributed system which is composed of tiny, low-cost and battery-operated web applications that collaborate together for the purpose of achieving certain task such as environment monitoring and web projects[3].

In the literature, the exploiting problem in IOT has been addressed either as object exploiting or Agile software development[1]. While area exploiting protocols are designed to maximize the area of the sensing field that could be covered, object exploiting, on the other hand, assumes that the sensing field is divided into objects. One of the key points in the design stage of an IOT that is related to the sensing attribute is the exploiting of the sensing field. Therefore, the main objective of the object exploiting protocols is to maximize the number of objects that could be covered in the field.

The method uses optimization in order to determine the minimum number of moving web applications that should be used in addition to the previously deployed non-moving web applications such that the exploiting of the monitored area is maximized. In this study, a method that exploits the movements of some web applications for eliminating the weakness was proposed, which would be formulated after the original distribution of the web applications.

In the first method, if all web applications are moving, then an efficient model should be designed such that the exploiting is maximized while, at the same time the moving cost of the moving web applications is minimized. In this case, the mobility feature of the web applications can be utilized in order to maximize the exploiting. In free distribution, the weakness formulation problem might be reduced or eliminated after original distribution using one of two methods.

One of the models that can be employed is an optimization model in which is used to find an optimal solution for web applications[4]. A little study in the field of IOT has used and employed optimization to search for an optimal number of web applications that can be added after the original web application distribution in order to maximize the exploiting. In the second method, if the web applications are mixed in which some of the web applications are non-moving and the other are moving, an efficient model should be employed in order to find the number and positions of the moving web applications that should be added after the original distribution of the non-moving web applications.

MATERIALS AND METHODS

Optimization method: In the initialization process, n mixtures are created as the first generation of solutions. After the initialization, each mixture is estimated using the fitness function. In general, an optimization model has four stages: Population initialization, evaluation of fitness, reproduction and termination. Initialization is the process of creating original free solutions, which can be done by setting genes to free values.

After selecting the mixtures, a crossover operation is performed by selecting a free point in mixtures and exchanging genes after this point. Crossover may be stuck in local optima. However, less fitness members will have also a chance to be selected. The selection step can be implemented by many mechanisms such as the wheel method. This selection will be performed on two mixtures to reproduce two new mixtures each time. To overcome this problem, a tie breaker is needed which can be achieved by using mutation operation, where a gene is selected freely and its value is changed. Reproduction process has four steps: Selection, cross-over, mutation and accepting the solution. In the selection step, the fittest members in the current population are selected in order to reproduce new solutions.

Many factors should be taken into consideration when the optimization model is used. The first factor is the representation of mixture and genes because bad representation may result in slower convergence. The best generated solutions will be added to the next iteration while, the bad solutions will be rejected. While the model iterates its solutions, these solutions are improved up to a point where converge to a near optimal solution is achieved. An optimization model is used to search for near optimal solutions when no deterministic method exists or if the deterministic method is computationally complex. Optimization is a population based model (i.e., it generates multiple solutions each iteration). The number of solutions per iteration is called population size. Each solution is represented as a mixture and each mixture is built up from genes. For an optimization model of population size n, it starts with n free solutions. Then it chooses the best member solutions for mating to generate new solutions. The third factor is how to find a fitness function (i.e., a method to evaluate the solutions) in order to accept or reject the solutions and how to select the best members for mating. Another important factor is the mechanism of producing new solutions from the old ones. The most popular mechanisms are crossover and mutation.

The most widely used stopping criterion is the number of iterations. When a predefined number of iterations are satisfied, the optimization model is terminated. Termination is the last step in the optimization model. Usually, the iteration of the optimization model is stopped when a certain criterion is met.

The final step is accepting these two mixtures to be in the new population. Typically, the new mixtures are accepted if they are better than their parents. A widely used representation for genes is bits where each gene is represented by a bit. In this case, mutation is done by flipping a bit freely in the mixture. After crossover and mutation, two new mixtures are reproduced.

Related work: The obtained solutions were analyzed for better distribution in the region of interest. Recently, a biogeography-based optimization model was proposed to maximize the exploiting area of the network. Mendes and Mosley[5], a distributed protocol was proposed that considers the different sensing capabilities of the web applications using realistic sensing exploiting model. In this protocol, the non-moving web applications determine the uncovered areas using a probabilistic exploiting model and the moving web applications move accordingly using virtual force model. In Hu[6], several techniques and complexities methods were proposed based on virtual force model and particle swarm optimization. On the other hand, several studies have considered both non-moving and moving web applications in IOT. Mendes and Mosley[5], a bidding protocol was proposed, in which the non-moving web applications are utilized as bidders and a number of moving web applications move accordingly to satisfy the exploiting requirements.

Polygon diagrams were used to partition the field into cells and an optimization model was then applied to determine the best positions for k extra moving web applications that maximize the area exploiting inside each cell. A force-based optimization model was proposed, in which the moving web applications utilize the sum of the forces used by the neighbors to choose their direction. The base station determines, where the moving web applications can move to maximize the exploiting and minimize the travelled distance. A cluster based IOT was considered and an optimization model was used to find the best positions for the cluster heads that cover the maximum number of web applications and hence maximizing the area exploiting. Optimization models have also been used to solve the problem of optimal web application distribution. While most of the proposed solutions have focused on deterministic web application distribution, few studies have been done in case of free web application distribution. In free distribution, optimization models are applied to determine near optimal positions for extra moving web applications in order to maximize the exploiting. Unlike the above-mentioned optimization models, this study proposes an optimization model that finds the minimum number of extra moving web applications and the best positions for these web applications in order to maximize the overall exploiting.

Hooi *et al.*[7], a potential field-based method was proposed, in which a repelled force is generated between the obstacles and web applications and among the web applications themselves, in order to evenly distribute the web applications in the field. A virtual force model was proposed in Kumari and Pushkar[8] that uses both pulling and pushing force among the web applications. Cagley[9], simulated annealing was used to find near optimal solutions for web applications to place that maximize the exploiting of the area of interest. Several research works have addressed the web application distribution problem to achieve maximum exploiting in IOT. For moving node networks, several methods have been proposed. Polygon diagrams were used in Reifer[10] to find the uncovered areas and determine the positions, where the web applications can move. For free web application distribution, these studies considered IOT that consist of estimating web applications[9,10] or that contain both non-moving and moving web applications[5,6,11].

Proposed method: It was present the proposed method. The network assumptions and exploiting model were 1st present and then discuss the optimization-based method.

Exploiting model: It was assumed that each node web application with a sensing radius r can cover an area of circular shape. It was also assumed that an object O_j can be detected by node S_i if O_j is within the sensing range of S_i. This can be represented using the binary model of node detection which is given by Eq. 1:

$$\text{Coverage (S)} = \begin{cases} 1, & D(S_i, O_j) \leq r \\ 0, & D(S_i, O_j) > r \end{cases} \quad (1)$$

where, D is the distance between the object being sensed O_j and the node web application S_i. The exploiting function exploiting (S) equals 1 when the object can be covered or sensed, otherwise it equals 0.

Network assumptions: It was also assumed that few moving web applications are available and can be used to repair the exploiting weakness after original distribution of the non-moving web applications. It was assumed that the web applications are freely deployed and equipped with GPS and the base station web application position is non-moving. Furthermore, the number of web applications that are originally deployed equals the number of web applications that are required to achieve full exploiting as if these web applications were deterministically deployed.

Optimization-based method: Then, the base station will run the optimization after gathering the positions of the non-moving web applications in order to determine the number and positions of the moving web applications as follow:

Optimization operators: The fitness ratio calculated in Eq. 3 as a measure for ranking the mixtures was used and then performing parent selection according to the ratio participated by each mixture in the fitness function. The mixtures with higher ranking have a higher probability to become parents and perform reproduction of new individuals than others with smaller ranking. This property must be maintained in order to increase the improvement ratio at each generation of the optimization. With high probability, a crossover operation is performed between a pair of parent mixtures in order to create two offspring. In the crossover operation, the crossover point is chosen freely where two parent mixtures are selected and then the mixture parts are exchanged after that point. The sequence of selection and crossover operations may lead to a state, where all mixtures are identical and thus the model stops creating new individuals. This may prevent the average fitness improvement and thus, trapping into a local optimum. To solve this problem, a mutation operation with low probability is applied to toggle the freely selected gene on the mixtures.

Termination condition: The termination condition of the applied model in terms of the number of generations were defined and exploiting ratio perspectives. That is the model terminates either when the required network exploiting ratio is reached or when the model reaches the specified number of generations.

Fitness function: Optimization, an objective function is defined in order to evaluate the fitness of each solution to the corresponding objective. The formulation of the objective or fitness depends on the problem characteristics. The fitness function is used in order to choose the best fittest mixtures for the purpose of reproduction of the next generated solutions by the optimization. The fitness function in our model defines the mutually exclusive exploiting ratio of each mixture. That is the fitness function calculates the maximum number of covered objects by each moving web application if and only if these objects are uncovered by other moving or non-moving web applications. This property of the fitness function prevents the overlapping redundancy among the exploiting regions of the deployed moving web applications and forces each moving web application to cover only a distinct region. The fitness function is given by Eq. 2:

$$F(M_{Si.}) = \begin{cases} F(M_{Si})+1, & D(M_{Si}, O_j) \leq r \\ \text{and} & O_J \notin \{S_C, F(M_{S/i})\} \\ F(M_{Si}), & \text{Otherwise} \end{cases} \quad (2)$$

where, $F(M_{Si})$ is the fitness of moving web application i (M_{Si}) which calculates the exploiting as a function of the objects it covered given that the object O_j is not covered by any non-moving web application or other moving web applications. In Eq. 2, S_c is the exploiting of originally deployed non-moving web applications and $F(M_{S/i})$ is the exploiting of any moving web application except moving web application i.

To choose the fittest species for mating in the next generation, a fitness ratio for each moving web application as a function of its exploiting was defined and the total number of objects which is given by Eq. 3:

$$\text{Fitness ratio} = \left[\frac{F(M_{Si})}{\sum O_j}\right]\% \quad (3)$$

A function that measures the total exploiting of the network at each optimization generation was also defined. This function is defined as an accumulation of the exploiting of the non-moving web applications and the generated moving web applications and is given by Eq. 4:

$$\text{Accumulated coverage} = S_C + \sum_{i-1}^{m} F(M_{Si}) \qquad (4)$$

Mixture modeling: Each mixture, as a solution in the optimization represents the position of a potential moving node web application in the sensing field modeled as (X, Y) point. The gens of each mixture represent a binary digit that resembles the value of the position on the X and Y axis. For example, in order to represent a moving web application mapped to position (30, 40).

The size of the mixture population is selected based on two factors: The area of the sensing field and the original configuration of the network. For instance, if the area of the sensing field is (50×50 m) the sensing radius of each web application is 8 m and the number of deployed non-moving web applications is 13 (i.e., $50^2/(\pi.8^2)\approx 13$), then the proposed optimization will start with population of 13 freely generated mixtures. Note that value 13 is selected in this case based on the assumption that 13 web applications would cover the entire field as if they were deterministically deployed.

The main objective of employing the optimization in the method is to maximize the exploiting by reducing or eliminating the weakness that are formulated after original distribution of the non-moving web applications. Assume that Si non-moving web applications are deployed freely over a sensing field and all the web applications have the same sensing range which is represented as a circle with radius r.

Performance of the proposed model: The performance of the proposed model is estimated in terms of the amount of exploiting (exploiting ratio), degree of exploiting (k-exploiting) and number of extra moving web applications. Moreover, the effect of the number of freely deployed non-moving web applications and the sensing ranges on exploiting and number of extra moving web applications were investigated.

Two simulation experiments were conducted for performance evaluation. In the simulation environment, it was assumed that the web applications were freely deployed and the objects were uniformly located in a 210×210 m node field. In the first experiment, the number of deployed non-moving web applications varies from 100-210 to cover 628 objects, whereas the sensing ranges of all web applications are fixed to 13 m. In the second experiment, the number of deployed non-moving web applications is fixed to 100, while the sensing ranges vary from 10-20 m. In each experiment, the exploiting ratio, k-exploiting and number of extra moving web applications were measured before and after applying the proposed model.

RESULTS

Figure 1 shows the exploiting ratio when the non-moving web applications are freely deployed and after adding the moving web applications to the network. As shown, the exploiting ratio increases as the number of deployed non-moving web applications increases.

Figure 2 shows the k-exploiting when the non-moving web applications are freely deployed and after adding the moving web applications to the network. For both cases, it is shown that as the number of web applications increases, the k-exploiting increases.

The number of extra moving web applications versus the number of freely deployed non-moving web applications is

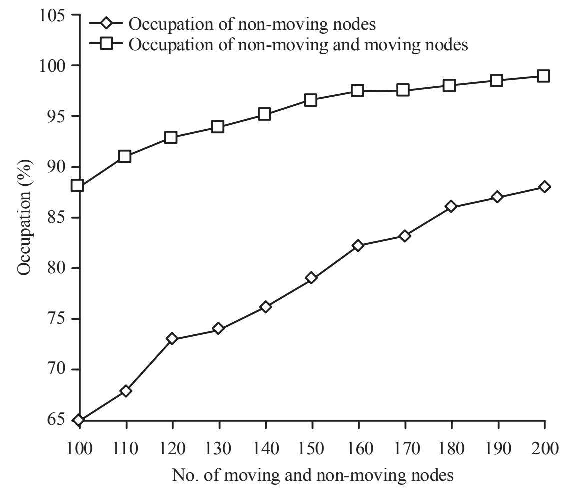

Fig. 1: Comparison of occupation ratio for different number of deployed nodes

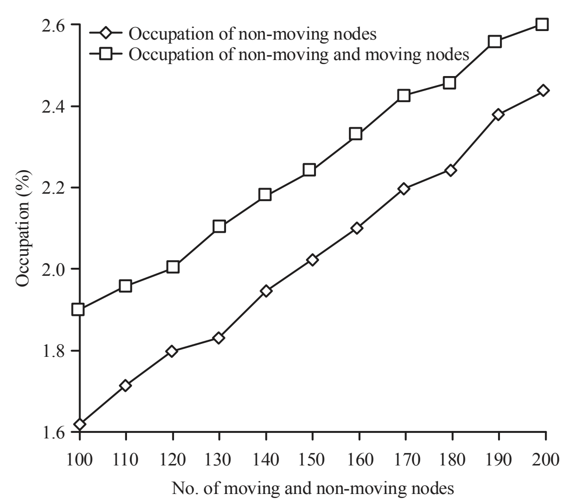

Fig. 2: Comparison of k-occupation ratio for different number of deployed nodes

shown, the number of moving web applications decreases as the number of non-moving web applications increases. This is because more objects would be covered as the number of non-moving web applications increases and hence less moving web applications would be added to increase the exploiting ratio.

It is shown that the exploiting ratio increases as the sensing radii of the deployed web applications increase, since web applications with larger sensing range can cover more objects than that with smaller range. The exploiting of the non-moving web applications along with the extra moving web applications clearly outperforms the case of free distribution of the non-moving web applications as the extra moving web applications are located into regions, where objects are not covered by the non-moving web applications.

As shown, the k-exploiting increases as the sensing radii of the deployed web applications increase. This is because the exploiting among web applications with large sensing range is very likely to overlap and hence, more objects would be covered by multiple web applications.

It is shown that the number of moving web applications decreases as the sensing radii of the web applications increase. This is because more objects would be covered as the sensing range of the non-moving web applications increases and hence, less moving web applications would be added to increase the exploiting ratio.

DISCUSSION

In IOT where all web applications are non-moving, the area of the sensing field and the number of web applications are small, exploiting can be maximized by manually deploying extra web applications to the originally deployed ones. Some scholars believe that human intervention is not possible in large scale IOT, free distribution is the only choice[4,7,11]. It was believe that an efficient model can maximize the covered area or objects should be employed through human intervention to overcome the problem of weakness formulation after original distribution of the web applications[1-3]. According to the application, the web applications might be non-moving, moving or mixed in which some of the web applications are non-moving and the others are moving.

The method uses optimization in order to determine the minimum number of moving web applications that should be used in addition to the previously deployed non-moving web applications such that the exploiting of the monitored area is maximized. In this study, a method that exploits the movements of some web applications for eliminating the weakness was proposed, which would be formulated after the original distribution of the web applications.

The purpose of the study is to present an optimization method that searches for an optimal solution to the exploiting weakness problem. According to recent research problem, such as optimization method, the minimum number and the best positions of web applications, exploiting object exploiting have discussed, then got the conclusion: (1) The proposed optimization method was performed to determine the minimum number and the best positions of the moving web applications. (2) Internet of things is composed of both non-moving and moving web application, the free distribution of non-moving web applications may cause exploiting weakness in the sensing field. (3) The non-moving web applications in both cases are freely deployed and it is very likely that the exploiting among these web applications is over-lapped, hence the objects would be covered by more web applications as the number of non-moving web applications increases.

CONCLUSION

This study presents an optimization model to find an optimal solution to the exploiting weakness problem caused by free distribution of non-moving web applications in internet of things. The performance of proposed model was estimated in terms of the exploiting ratio, k-exploiting and the number of extra moving web applications using different numbers of non-moving web applications and various sensing ranges. The simulation results showed that the optimization model can maximize the exploiting of the sensing field by finding the minimum number of extra moving web applications and their best positions in the field.

ACKNOWLEDGMENT

This study was supported by the Xi-an Science and Technology Plan Project CXY1443WL17. This study is supported by Natural Science Fund Project in Guang Dong province (2015A030313671) and the major Science and Technology Projects in Guangzhou city (201605080826025).

REFERENCES

1. Zia, Z.K., Z.K. Zia, S.K. Tipu and S.K. Zia, 2012. An effort estimation model for agile software development. Adv. Comput. Sci. Applic., 2: 314-324.
2. Ochoa, S.F., M.C. Bastarrica and G. Parra, 2003. Estimating the development effort of web projects in Chile. Proceedings of the 1st Latin American Web Congress, November 10-12, 2003, Sanitago, Chile, pp: 114-122.

3. Lazic, L. and N.E. Mastorakis, 2010. Two novel effort estimation models based on quality metrics in web projects. WSEAS Trans. Inform. Sci. Applic., 7: 923-934.
4. Ruhe, M., R. Jeffery and I. Wieczorek, 2003. Cost estimation for web applications. Proceedings of the 25th International Conference on Software Engineering, May 3-10, 2003, Germany, pp: 285-294.
5. Mendes, E. and N. Mosley, 2005. Web Cost Estimation: An Introduction. In: Web Engineering: Principles and Techniques, Suh, W. (Ed.). Idea Group Inc., Canada, ISBN: 9781591404330, pp: 182-190.
6. Hu, R., 2012. Channel access controlling in wireless integrated information network using smart grid system. Applied Mathe. Inform. Sci., 6: 813-820.
7. Hooi, T.C., Y. Yusoff and Z. Hassan, 2008. Comparative study on applicability of WEBMO in web application cost estimation within Klang valley in Malaysia. Proceedings of the IEEE 8th International Conference on Computer and Information Technology Workshops, July 8-11, 2008, Sydney, pp: 116-121.
8. Kumari, S. and S. Pushkar, 2013. Comparison and analysis of different software cost estimation methods. Int. J. Adv. Comput. Sci. Applic., 4: 153-157.
9. Cagley, Jr.T.M., 1994. Agile estimation using functional methods. Vice President David Consulting Group. http://www.softwarevalue.com/media/1084244/agile-estimation-using-functional-metrics.pdf
10. Reifer, D.J., 2000. Web development: Estimating quick-to-market software. IEEE Software, 17: 57-64.
11. Hu, R., 2012. Stability analysis of wireless integrated information network service via data stream methods. Applied Mathe. Inform. Sci., 3: 793-798.